International Journal of Engineering Inventions
ISSN: 2278-7461, www.ijeijournal.com
Volume 1, Issue 2 (September 2012) PP: 01-07

Resolving Cloud Application Migration Issues

C.Kishor Kumar Reddy[1], P.R Anisha[2], B.Mounika[3], V.Tejaswini[4]
[1234]*Department of CSE, Vardhaman College of Engineering, Shamshabad*

Abstract--Cloud computing is a rapidly emerging technology which involves deployment of various services like software, web services and virtualized infrastructure, as a product on public, private or hybrid clouds on lease basis. Cloud storage can be an attractive means of outsourcing the day-to-day management of data, but ultimately the responsibility and liability for that data falls on the company that owns the data, not the hosting provider. In part one of this look at cloud application migration, we discussed how cloud providers, through the selection of hypervisors and networking, affect the capability to migrate applications. In part two, we will talk about how appropriate architectures for cloud applications and open standards can reduce the difficulty in migrating applications across cloud environments. This paper highlights the Tools to facilitate application migration in clouds. This paper outlines architectures and indicates when each is most appropriate based on system needs and cloud provider capabilities.

Keywords: Virtualized Infrastructure, Public Cloud, Private Cloud, Hybrid Cloud, Cloud Storage, Application Migration, Hypervisors.

I. INTRODUCTION

Although the term "Cloud Computing" is based on a collection of many old and few new concepts in several research fields like Service-Oriented Architectures (SOA), distributed and grid computing as well as virtualization, it has created much interest in the last few years. This was a result of its huge potential for substantiating other technological advances while presenting a superior utilitarian advantage over the currently under-utilized resources deployed at data centers. In this sense, cloud computing can be considered a new computing paradigm that allows users to temporary utilize computing infrastructure over the network, supplied as a service by the cloud-provider at possibly one or more levels of abstraction. Consequently, several business models rapidly evolved to harness this technology by providing software applications, programming platforms, data-storage, computing infrastructure and hardware as services. While they refer to the core cloud computing services, their inter-relations have been ambiguous and the feasibility of enabling their inter-operability has been debatable. Furthermore, each cloud computing service has a distinct interface and employs a different access protocol. A unified interface to provide integrated access to cloud computing services is nonexistent, although portals and gateways can provide this unified web-bas based user interface.

In part one of this look at cloud_application_migration, we discussed how cloud providers, through the selection of hypervisors and networking, affect the capability to migrate applications. In part two, we will talk about how appropriate architectures_for_cloud_applications and open standards can reduce the difficulty in migrating applications across cloud environments.

A good deal of time and money in the IT industry has been spent on trying to make applications portable. Not surprising, the goal around migrating applications among clouds is to somehow make applications more cloud-portable. This can be done in at least three ways:

Architect applications to increase cloud portability,Develop open standards for clouds.
Find tools that move applications around clouds without requiring changes.

Most of today's large, old monolithic applications are not portable and must be rebuilt in order to fit the target environment. There are other applications that require special hardware, reducing their portability, and even many of the newer applications being built today are not very portable, certainly not cloud portable.

II. APPLICATION ARCHITECTURES AND THE CLOUD

Numerous cloud experts have indicated how important an application's architecture reflects on the ability to move it from one cloud to another. Appropriate_cloud_application_architectures are part of the solution to cloud interoperability, and existing applications may need to be re-architected to facilitate migration. The key is trying to architect applications that reduce or eliminate the number of difficult-to-resolve dependencies between the application stack and the capabilities provided by the cloud service provider.

Bernard Golden, CEO of HyperStratus, has noted that, to exploit the flexibility of a cloud environment, you need to understand which application architectures are properly structured to operate in a cloud, the kinds of applications and data that run well in cloud environments, data backup needs and system workloads.

There are at least three cloud application architectures in play today:

Traditional application architectures (such as three-tier architectures) that are designed for stable demand rather than large variations in load. They do not require an architecture that can scale up or down.

Synchronous application architectures, where end-user interaction is the primary focus. Typically, large numbers of users may be pounding on a Web application in a short time period and could overwhelm the application and system.

Asynchronous application architectures, which are essentially all batch applications that do not support end-user interaction. They work on sets of data, extracting and inserting data into databases. Cloud computing offers scalability of server resources, allowing an otherwise long running asynchronous job to be dispersed over several servers to share the processing load.

Platform as a Service (PaaS) providers provide tools for developing applications and an environment for running these applications. To deliver an application with a PaaS platform, you develop and deploy it on the platform; this is the way Google App Engine works. You can only deploy App Engine applications on Google services, but cloud application platforms such as the Appistry CloudIO Platform allow for in-house private cloud deployment as well as deployment on public cloud infrastructures such as Amazon EC2.

Where the application is developed and where it is to be run are factors that feed into the application architecture. For example, if you develop in a private cloud with no multi-tenancy, will this application run in target clouds where multi-tenancy is prevalent? Integrating new applications with existing ones can be a key part of application development. If integration involves working with cloud providers, it is difficult because cloud providers do not typically have open access into their infrastructures, applications and integration platforms.

Older applications that depend on specific pieces of hardware -- meaning they'll want to see a certain type of network controller or disk -- are trouble as well. The cloud provider is not likely to have picked these older pieces of hardware for inclusion in its infrastructure.

In your efforts to migrate applications, you may decide to start working with a cloud provider template where the provider gives you an operating system, such as CentOS or a Red Hat Enterprise Linux template. You'll then try to put your applications on it, fixing up the things that are mismatched between the source application environment and the target environment. The real challenge is that this approach becomes an unknown process, complete with a lot of workarounds and changes.

As you move through a chain of events, fixing problems as you go, you are really rewriting your application. Hopefully you won't have to rewrite it all, but you will surely change configurations and other things. You are then left with a fundamentally different application. This could be good or bad, but either way you'll have at least two versions of your application -- the data center version and the cloud version.

If moving an application back and forth between your data center and a cloud (or from one cloud to another cloud) results in two different versions of the application, you are now managing a collection of apps. As you fix and encounter problems, you'll have to work across however many versions of the application that you have created.

Open cloud standards are considered the eventual solution to issues around application migration and cloud interoperability. We view cloud standards as a collection; this one starts at the low level with something like OVF (Open Virtualization Format) that gives you a universal language for describing the metadata and configuration parameters of virtual machines. At the next level, something that would describe the environment -- the connectivity between virtual machines -- would be useful. This would give you the networking between the virtual machines *and* the functions and scale of the environment in which the virtual machines operate.

How to build an application for the cloud

Applications interfere with cloud computing adoption

It is unlikely that we will see cloud standards being adopted this year or next year, for reasons that include ongoing innovation. Vendors such as VMware would love to just say, "We will do the whole black box thing for you: buy our stuff and you can put up a cloud and offer it to your customers." The cloud providers are not thrilled with this idea because they want to differentiate their services. They don’t want to go the route of standards if clouds are then driven to commodities. If and when we have standards, there will almost certainly be a problem with how cloud providers do or offer unique things on top of standards.

John Considine, the CTO of CloudSwitch, notes that for a cloud provider, a standard provides customers with what they want and provides a guideline for how cloud is implemented. In the case of the VMware vCloud API -- which has been submitted to the DMTF for ratification as an open standard for cloud APIs -- VMware dictates how cloud environments are configured and accessed with respect to things like definition of resources and catalogs of virtual machines. These "mandates" have a direct impact on how a provider implements its cloud.

What are some hints for architecting cloud applications? One suggestion is to design the application and its supporting stack components not to rely on the operating system and the infrastructure. The more you do this, the better off you will be with respect to interoperability and application migration. If you can use mature fourth-generation languages or interpretive systems to build applications, then you will also have a better chance for interoperability.

The problem you might run into is not getting the performance and/or the functionality you need. In addition, you may have to avoid certain performance and capability benefits that could be available with hypervisor tools or from the specifics of an operating system. You also might have to go for a generic operation of your application with min-set functionality to make it portable from cloud to cloud.

What kind of existing applications are good candidates for running in the cloud? The more generic and higher level the application is, the greater your chances of moving it from cloud to cloud. One of the cloud's weakest areas is in needing total control over the operating system. If you are running an old version of Linux or Windows, then you are probably in trouble; most public clouds do not support older versions of operating systems. This is a dating problem, as applications written before a certain date are not easily movable.

Migrating applications among clouds is not easy. But open standards for cloud computing, when they appear, and the advent of tools such as CloudSwitch and Racemi will ease the difficulty and make hybrid clouds more of a reality.

III. HOW PROVIDERS AFFECT CLOUD APPLICATION MIGRATION

In an attempt to reduce lock-in, improve cloud interoperability and ultimately choose the best option for their enterprises, more than a few cloud computing users have been clamoring for the ability to seamlessly migrate applications from cloud to cloud. Unfortunately, there's more to application migration than simply moving an application into a new cloud.

To date, cloud application migration in clouds has focused on moving apps back and forth between a virtualized data center or private cloud environment and public clouds such as Amazon Elastic Compute Cloud, Rackspace or Savvis. There is also a group of public cloud-oriented companies that are looking to move applications to private clouds or virtualized data centers to save money. Still others are interested in moving applications from one public cloud to another in a quest for better service-level agreements and/or performance at a lower cost.

What are some of the worries in moving an application from one environment to another?
Data movement and encryption, both in transit and when it reaches the target environment.

Setting up networking to maintain certain relationships in the source environment and preparing to connect into different network options provided by the target environment.

The application itself, which lives in an ecosystem surrounded by tools and processes. When the application is moved to a target cloud, you may have to re-architect it based on the components/resources that the target cloud provides.

Applications are being built in a number of ways using various Platform as a Service offerings, including Windows Azure, Google App Engine and Force.com. With a few exceptions for Windows Azure, the applications you create using most platforms are not very, if at all, portable. If you develop on Google App Engine, then you have to run on Google App Engine.

Public clouds, like Amazon, also allow you to build applications. This is similar to building in your data center or private cloud, but public clouds may place restrictions on resources and components you can use during development. This can make testing difficult and create issues when you try to move the application into production mode in your data center environment or to another cloud environment.

Applications built in data centers may or may not be easily moved to target cloud environments. A large number of applications use third-party software, such as database and productivity applications. Without access to source code, proprietary third-party applications may be difficult to move to clouds when changes are needed.
Why is migrating apps in the cloud so difficult? During traditional application development, we have complete control of everything we need: a physical server, an operating system, the amount of memory needed, the disk storage system, network configuration, patching and runtime systems such as Java.

But when server consolidation came along, our notion of environment for applications changed somewhat. We are given an application environment that we still control, sans the personal physical server. Hypervisors provide us with hardware and all other aspects of an environment, so that our operating system, middleware and applications still have all of the things that they want and need.

Application architectures for cloud computing environments
Microsoft brings rapid application development to the cloud

And when it comes to clouds, providers pick the operating system, management tools, the networking architecture, the storage system and the virtual machine configuration. This is the baseline with which you work, offering much less control of the environment for developing and deploying applications. If you want to move some of your applications to a cloud, then you have to make them work in an environment that will almost certainly be different from the one in which they were last deployed and/or developed.

The decisions made by the cloud provider affect what you can and cannot do within the cloud. For example, a cloud provider could decide to go with cheap disk drives and do big NFS shares or iSCSI shares to feed the virtualization layer, which sets a performance envelope, a protection level and a raw capacity for the storage for your virtual machine. You do not generally get a say in any of this; you must live with it.

Take a look at Amazon. The public cloud leader has decided that your virtual machine gets one network adapter with one interface. It also doesn't support layer 2 connectivity or broadcast and multi-cast. When you go into these types of environments, the design of your application is constrained by the cloud resources, in this case networking, that you're allowed to access.

Another example further illustrates how functions at various levels can affect your ability to move applications to clouds and from cloud to cloud. You have chosen MySQL as the database system for your application. One of the functions you can perform with MySQL is database replication -- two instances of a database kept in synch. The replication process in MySQL utilizes multi-cast as part of low-level Ethernet to communicate between the two database instances. If you want to run your application on Amazon *and* replicate a database, you have a problem: Amazon does not support multi-cast in its networking layers.

What do you do? You could use a different database in your application, or you could implement the replication function differently, but that would require messing with MySQL or handling it through the application. Either way, if you're thinking about interoperability and moving applications from cloud to cloud, you'll also have to think about top-to-bottom integration of your entire software stack into a target cloud environment.

The challenge is that you may not know all of the dependencies between the components of your application stack and elements of the cloud itself. There is a lot of value to extract from the use of clouds. but if you have to expend a lot of energy and if it creates a burden for maintenance and support, you greatly diminish whatever gains you get.

IV. TOOLS TO FACILITATE APPLICATION MIGRATION IN CLOUDS

Given the differences in environments, some customers may not want to go through the oft-difficult process of making an application work in a target cloud. But if we can give the virtual machine in the new cloud exactly what it wants to see -- independent of the hypervisor, the cloud environment it is on -- then application migration is easier. This is what products like CloudSwitch and Racemi offer.

Applications built in data centers may or may not be easily moved to target cloud environments.

CloudSwitch facilitates multi-tier application migration in the cloud with its Cloud Isolation Technology, which is a virtualization technology layer that automatically runs on top of the cloud provider's hypervisor and beneath the end user's operating system. The virtualization layer feeds the virtual machine exactly what it wants to see -- without requiring anything special from the cloud provider -- and runs on behalf of the customer to protect and isolate an environment in the cloud. Applications do not have to be modified when using CloudSwitch; it maps the application so that it seems to be running within the target cloud environment while actually maintaining the same exact configuration as in the source environment.

Racemi takes a different approach to migrating applications than CloudSwitch. It involves capturing a server, physical or virtual, in one environment (data center or cloud) and then deploying it in a target environment (data center or cloud). An important component of the Racemi application migration offering is a management appliance that has access to both the captured server environment and the target server environment and begins a mapping process between the two. Once this mapping process is completed, the capturing-deploying activity is complete and the application has been migrated to the target environment.

Migrating applications can be quite a big issue in hybrid cloud and a little less of an issue when you just want to move an application from your data center to a public cloud. You may, however, still encounter some of the problems discussed above.

Application migration involves more than just dealing with potentially incompatible cloud application programming interfaces. There are potential issues at each level of an application stack, as your two clouds are very likely to have differences in hypervisors, operating systems, storage and network configurations, and drivers.

When you are creating cloud environments with the intention of moving applications around, you need to perform a thorough investigation of the differences in the cloud environments and look at your application architectures to determine if they are reasonable fits. The second part of this piece on cloud application migration will offer up some possible solutions to difficult migration problems.

V. CLOUD APPLICATION

A cloud application (or cloud app) is an application program that functions in the cloud, with some characteristics of a pure desktop app and some characteristics of a pure Web app. A desktop app resides entirely on a single device at the user's location (it doesn't necessarily have to be a desktop computer). A Web app is stored entirely on a remote server and is delivered over the Internet through a browser interface.

Like desktop apps, cloud apps can provide fast responsiveness and can work offline. Like web apps, cloud apps need not permanently reside on the local device, but they can be easily updated online. Cloud apps are therefore under the user's constant control, yet they need not always consume storage space on the user's computer or communications device. Assuming that the user has a reasonably fast Internet connection, a well-written cloud app offers all the interactivity of a desktop app along with the portability of a Web app.

If you have a cloud app, it can be used by anyone with a Web browser and a communications device that can connect to the Internet. While tools exist and can be modified in the cloud, the actual user interface exists on the local device. The user can cache data locally, enabling full offline mode when desired. A cloud app, unlike a Web app, can be used on board an aircraft or in any other sensitive situation where wireless devices are not allowed, because the app will function even when the Internet connection is disabled. In addition, cloud apps can provide some functionality even when no Internet connection is available for extended periods (while camping in a remote wilderness, for example).

Cloud apps have become popular among people who share content on the Internet. Linebreak S.L., based in Spain, offers a cloud app named (appropriately enough) "CloudApp," which allows subscribers to share files, images, links, music, and videos. Amazon Web Services offers an "AppStore" that facilitates quick and easy deployment of programs and applications stored in the cloud. Google offers a solution called "AppEngine" that allows users to develop and run their own applications on Google's infrastructure. Google also offers a popular calendar (scheduling) cloud app.

Application architectures for cloud computing environments

The ways that organizations use applications, and the applications themselves, are evolving. Applications nowadays work with increasing amounts of data while suffering from inconsistent loads and use patterns. In short, they're well-suited for cloud computing environments.

In this webcast, Bernard Golden, CEO of cloud consulting firm HyperStratus, describes the key principles of application architectures in a cloud computing environment. He also discusses the limitations of running applications in the cloud, including application integration issues, software licensing and security.

Choosing an application architecture for the cloud

For most of us, building applications to run in a cloud environment is a new ball of wax. But to exploit the flexibility of a cloud environment, you need to understand which application architectures are properly structured to operate in the cloud, the kinds of applications and data that run well in cloud environments, data backup needs and system workloads. There are three architectural choices for cloud environments:

Traditional architecture;
asynchronous application architecture (i.e., one focused on information processing, not end-user interaction); and
Synchronous application architecture.

As part of your system design, you need to research and design your application appropriately for the particular environment offered by your cloud provider. Cloud environments are not all created equal: They offer different mechanisms to implement applications. Choosing from the major Platform as a Service providers Platform as a Service is the preferred cloud computing approach for many -- if not most -- organizations, as it frees software developers and IT operatives from infrastructure management chores, security concerns and licensing issues.

Amazon's new Elastic Beanstalk service, albeit in a beta version, enables Amazon Web Services to join Google and Microsoft in forming a triumvirate of top-tier Platform as a Service (PaaS) providers. These organizations readily pass most governance standards established by upper-level management and boards of directors for IT vendors, such as the following:

Weighing the cloud computing standards dilemma

Many IT pros view the lack of cloud computing standards as a potential roadblock to adoption, stemming from cloud provider lock-in fears and the inability to move virtual machines and data from cloud to cloud. Today, there is a single cloud standard -- the Open Virtualization Format (OVF), pioneered by VMware for facilitating the mobility of virtual machines -- but it alone does not solve the cloud interoperability issue. The vendors with the best shot at providing de facto cloud API standards are VMware and Amazon. What users want is a cloud application programming interface (API) like the network API, TCP/IP, one that’s implemented in all cloud products and services and promotes transparent interoperability. This would increase the confidence of prospective public cloud adopters, as they’d be able to leave their providers whenever they want. It would also eliminate the belief that it’s easy to get into the cloud but difficult to get out. However, Forrester analyst James Staten says he believes that a common cloud API is *way off* in the future. He sees the push for standards as too far ahead of where the market is: “There is no compelling reason to comply; not enough enterprise users have established cloud computing initiatives.”

Organizations presently pushing for cloud standards

There are a number of organizations that ratify proposals for open standards and others that develop guidelines and provide information to those interested in cloud computing. Some of the more important ones include:

The Distributed Management Task Force (DMTF) develops cloud interoperability and security standards. The DTMF created the Open Cloud Standards Incubator (OCSI) in 2009 to address the need for open management standards for cloud computing. The mission of the National Institute of Standards and Technology (NIST) is to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security.The Open Cloud Consortium (OCC) is a member-driven organization that develops reference implementations, benchmarks and standards for cloud computing. The Open Grid Forum (OGF) is an open community committed to driving the rapid evolution and adoption of applied distributed computing. OGF accomplishes its work through open forums that build the community, explore trends, share best practices and consolidate these best practices into standards. OGF has launched the Open Cloud Computing Interface Working Group to deliver an open community, consensus-driven API, targeting cloud infrastructures. The Storage Networking Industry Association (SNIA) has adopted the role of industry catalyst for the development of storage specifications and technologies, global standards, and storage education. The Cloud Security Alliance (CSA) publishes guidelines for secure cloud computing, and the Cloud Computing Interoperability Forum (CCIF) is a vendor-neutral, open community of technology advocates and consumers dedicated to driving the rapid adoption of global cloud computing services.

VI. THE CLOUD STANDARDS AVAILABLE TODAY

Various proprietary and open APIs have been proposed to provide interoperability among Infrastructure as a Service (IaaS). As far as I know, however, the first and only cloud-oriented standard that has been ratified is the OVF, which was approved in September 2010 after three years of processing by the DMTF.

OVF’s open packaging and distribution format for virtual machines (VMs) gives customers and vendors some platform independence. It helps facilitate mobility, but it does not provide all of the independence needed for cloud interoperability. OVF lets vendors or enterprises package VMs together with applications and operating systems and calls to any other applications and hardware as needed. This meta data includes information about VM images, such as the number of CPUs and memory required and network configuration information.

Understanding cloud compliance issues
Security issues in cloud computing

Terremark guru talks enterprise cloud security issues As for the other prominent APIs, VMware announced the submission of its vCloud API to the DMTF in September 2009. The API is expected to enable consistent provisioning, management and service assurance of applications running in private and public clouds. The GoGrid API was submitted to the Open Grid Forum’s OCCI working group. This effort stalled because the GoGrid API failed to achieve any significant backing from the industry.

Oracle recently published its Oracle Cloud API on OTN (Oracle Technical Network) and submitted a subset of the API to the DMTG Open Cloud Standards Incubator group in June 2010. The Oracle Cloud API is basically the same as the

Sun Cloud API but with some refinements. This submitted proposal has not received much attention from the IT industry. Red Hat submitted the Deltacloud API to DMTF as a standard for cloud interoperability in August 2010. It is a set of open APIs that can be used to move cloud-based workloads among different private clouds and public cloud providers. Red Hat has contributed Deltacloud to the Apache Software Foundation as an incubator project. Deltacloud attempts to abstract the details of cloud provider cloud implementations so that an application or developer writing an application only has to call a single API to get a response regardless of the back-end cloud.

The Rackspace Cloud API has been open sourced and is included in OpenStack. And finally, the Amazon EC2 API is viewed by many as the de facto public cloud API standard. Vendors such as Eucalyptus Systems and OpenNebula implement much of the Amazon EC2 API. As far as we know, it has not been submitted to DMTF or any other open standards group.

VII. QUESTIONS TO POSE ON CLOUD STANDARDS

Because interoperability standards between cloud platforms are not yet in place, what should a prospective cloud adopter do? For starters, do not wait for interoperability standards to be ratified. In an environment of tremendous change where the potential benefits could be large, a better decision is to study up and make a choice. Try to determine which vendors have the best opportunities to turn their cloud APIs into de facto standards for private and public clouds. Be sure to ask a number of questions like these and then compare the answers with your needs for both the short and the long term:

Does the vendor have cloud APIs that appeal to customers and service providers, along with widespread acceptance in the cloud marketplace?

Has the vendor submitted one or more of its cloud APIs for ratification to DMTF or one of the other standards organizations?

Does the vendor have significant numbers of partners to promote and use its cloud APIs?

Does the vendor's cloud API promote interoperability between private and public clouds? What about between the vendor's cloud and another vendor's cloud?

Can the vendor provide a way to transfer your data out of its service?

What users want is a cloud API like TCP/IP, one that's implemented in all cloud products and services. If you woke up today and wanted to move data from one cloud to another, there is a good chance that you cannot. If you want to do this in the future, however, make sure that you don't architect your solution to take advantage of a vendor's proprietary features and services, such as the data store in Google App Engine or Amazon's SQS (Simple Queue Service). In my opinion, the vendors with the best shot at providing de facto cloud API standards are VMware and Amazon. Amazon is the 800-pound gorilla in the public cloud space, but VMware is trying to leverage its dominance in the virtualization software market to gain the lead in both the private and public cloud markets. If customers go heavily for internal clouds first, as many are predicting, VMware could become the de facto standard in the private cloud business. And with the vCloud Express offering and the vCloud API, VMware also has a good chance at challenging Amazon in the public cloud. Over a year ago, VMware CEO Paul Maritz said that more than 1,000 hosting providers have enlisted to help enable public clouds via vCloud Express. However, the vCloud API *could* be considered a lock-in API, unless other hypervisor technologies in the cloud beyond vSphere actually adopt or make use of it.

As for the other major player in cloud, Microsoft's strategy is aimed at its huge Windows-installed base. I don't expect Microsoft to make its APIs open, as its clouds don't have to interoperate with other clouds for it to be a successful vendor. Even Microsoft's own services, Hyper-V Cloud and Windows Azure, have limited interoperability. The reason: Microsoft believes that Azure is the wave of the future. If you are a large, cloud-hungry Windows user who wants to buy from a single vendor and doesn't object to lock-in (neither of which is advisable), choose Microsoft.

VIII. CONCLUSION

Cloud computing is a rapidly emerging technology which involves deployment of various services like software, web services and virtualized infrastructure, as a product on public, private or hybrid clouds on lease basis. As a part of cloud application migration, we discussed how cloud providers, through the selection of hypervisors and networking, affect the capability to migrate applications. In another part, we discussed about how appropriate architectures for cloud applications and open standards can reduce the difficulty in migrating applications across cloud environments. We highlighted the Tools to facilitate application migration in clouds. We highlighted outlines architectures and indicates when each is most appropriate based on system needs and cloud provider capabilities.

REFERENCES

1. Wayne Jansen and Timothy Grance. "Guidelines on Security and Privacy in Public Cloud Computing" NIST 800-144
2. Allan A. Friedman and Darrell M. West. "Privacy and Security in Cloud Computing " Issues in Technology Innovation. October 2010.
3. Siani Pearson. "Taking Account of Privacy when Designing Cloud Computing Services" HP Laboratories
4. The MITRE Corporation, "Common Vulnerability and Exposures (CVE)," http://cve.mitre.org/, Mar. 2011.
5. C. Gentry, "Fully homomorphic encryption using ideal lattices," in *STOC*, 2009.
6. C. Gentry, "Computing arbitrary functions of encrypted data," *Commun. ACM*, March 2010.
7. http://wattsupmeters.com/.
8. Amazon Case Studies. *http://aws.amazon.com/solutions/case-studies/*.

The author C. Kishor Kumar Reddy is a student, received his Bachelor Degree in the stream of Information Technology from Intellectural Institute of Technology, Anantapur(IIT-A), formerly known as MoulaAli College of Engineering and Technology in the year 2011. Currently He is pursing his M.Tech degree from Vardhaman College of Engineering in the stream of Computer Science. He has attended many international conferences and is awarded as Young Investigatoraward and is still publishing his papers in International Journals. His field of Interest Is Computer Networks and Cloud Computing.

The author P.R Anisha is a student, received her Bachelor Degree in the stream of Information Technology ,Hyderabad), from Maheshwara college of Engineering And Technology in the year 2011. Currently ,She is pursing her M.Tech degree from Vardhaman College of Engineering in the stream of Software Engineering. She has attended many international conferences and is still publishing her papers in International Journals. Her field of Interest Is Cloud Computing.